Projet : classifieur bayésien et perceptron multi classe

Master Pro — Traitement Statistique de l'Information

2008–2009

1 Objectifs

Le but de ce projet est de comparer deux approches de classification supervisée :

1. un classifieur bayésien paramétrique
2. un perceptron multi classe.

Les performances de ces deux classifieurs seront évaluées sur une base de données artificielles et sur deux problèmes de reconnaissance (identification d'iris et reconnaissance de caractères).

2 Évaluation

2.1 Travail demandé

Vous devrez effectuer les tâches suivantes :

- programmer en Octave les différentes fonctions décrites dans le reste du sujet ;
- fournir le résultat de votre meilleur système sur les données de test dans un fichier au format texte ;
- rédiger un rapport court (entre 5 et 10 pages) expliquant vos expériences et répondant aux questions posées. Le rapport est à rendre au secrétariat du Master Pro ; les sources et la version électronique du rapport sont à envoyer par mel aux adresses `claude.barras@limsi.fr` *et* `guillaume.wisniewski@limsi.fr` **au plus tard le 19 novembre 2008**.

2.2 Modalités d'évaluation

Les projets seront réalisés en binômes. Chaque binôme devra réaliser l'ensemble du projet. Le travail effectué sera évalué sur :

- Le rapport, accompagné d'un listing des programmes. Ce rapport devra détailler les contributions de chaque membre du binôme, justifier les choix effectués pour l'optimisation du système, commenter les matrices de confusion et les taux de reconnaissance du système et présenter les améliorations aussi bien programmées qu'envisagées. Les notes tiendront compte de la qualité de la présentation, mais aussi de l'initiative personnelle et de l'intérêt porté à l'expérimentation et à la recherche de meilleures solutions.
- Les programmes commentés, accompagnés d'un mode d'emploi. La lisibilité et l'efficacité du programme (en exploitant les spécificités d'octave) seront prises en compte.

Il est à noter que les membres d'un même binôme peuvent avoir des notes différentes si le travail n'est pas effectué équitablement.

3 Classifieurs étudiés

3.1 Classifieur gaussien

Le premier classifieur étudié est un classifieur implémentant la règle de décision bayèsienne. On modélisera les probabilité conditionnelles de chaque classe par des gaussiennes multi dimensionnelles.

1. Écrire une fonction d'apprentissage qui estime les paramètres nécessaires (probabilité à priori et probabilité conditionnelle $p(\mathbf{x}|\omega)$) à partir d'une base d'apprentissage fournie en paramètre.
2. Écrire une fonction de test qui retourne la performance du classifieur sur une base fournie en paramètre.

On se propose d'étudier les performances de ce classifieur sur une base de données artificielles afin d'évaluer l'impact du nombre de données d'apprentissage sur les performances en classification.

3. Écrire une fonction qui génère aléatoirement une base d'exemples de $\mathbb{R}^2$ selon deux gaussiennes.
4. Utiliser la fonction précédente pour construire une base de test et des bases d'apprentissage de différentes tailles.
5. Tracer la courbe représentant le nombre d'erreurs en fonction de la taille de l'ensemble d'apprentissage. On représentera également le nombre d'erreurs optimal. Que peut-on en déduire ?

3.2 Perceptron multi classe

Le deuxième classifieur étudié est une extension du perceptron aux problèmes multi classes. On adoptera pour cela la stratégie « un contre tous » qui consiste à :

- apprendre, pour chaque classe i, un perceptron binaire p_i capable de reconnaître les exemples de la classe i (en considérant les éléments de la classe i comme les exemples positifs et les éléments de toutes les autres classes comme les exemples négatifs)
- classer une observation $\mathbf{x}$ selon la règle de décision :

$$\mathbf{y} = \underset{i}{\text{argmax}}\, \mathbf{w_i} \cdot \mathbf{x}$$

  où $\mathbf{w_i}$ est le vecteur de paramètres du $\text{i}^{\text{ème}}$ perceptron.

6. Écrire une fonction apprenant les paramètres d'un perceptron binaire à partir d'une base d'apprentissage. On proposera une solution pour pouvoir traiter le cas de données non linéairement séparables.
7. Écrire une fonction généralisant le classifieur précédent aux problèmes multi classes.
8. Tester les deux fonctions précédentes dans le cas de données artificielles générées à partir de gaussiennes.

3.3 Comparaison des deux classifieurs sur des données réelles

Deux jeux de données sont fournis à l'url `http://www.limsi.fr/Individu/wisniews/enseignement/08-09_tsi_m2pro/index.html` : un jeu d'apprentissage et un jeu de test qui sont étiquetés (fichiers *.test et *.train), et un jeu d'évaluation (*.eval) correspondant à une suite d'exemples de classe inconnue. Les données sont représentées sous la forme de matrice octave (lisible par la fonction `load` d'octave). La classe de chaque exemple correspond à la dernière composante du vecteur, les classes des ensembles d'évaluation ne sont pas données.

9. Appliquer les deux classifieurs programmés précédemment à ces données. Comparer les résultats obtenus (après avoir choisi « intelligemment » la valeur des différents paramètres)
10. La matrice présentant les erreurs par classe reconnue en fonction de la classe réelle (appelée matrice de confusion) permet d'analyser plus finement le type des erreurs produites. Il vous est demandé de fournir

la matrice de confusion pour les données de développement, ainsi que le taux d'erreur par classe et le taux d'erreur global, sur vos meilleurs systèmes.

11. Vous devez également fournir le résultat de votre meilleur système sur les données d'évaluation dans un fichier au format texte (une étiquette par ligne).